CDF DATA ACCESS

Elizabeth Buckley–Geer
Von Ruden Review
January 26 1999

- Introduction
- Overall design strategy
- Logical data flow and data streams
- Analysis hardware design
- Data Access System design
- Components of Data Access System
- Production Farms
- Schedule and Manpower for CDF Data Access Joint Project
- Conclusions

Introduction – the good, the bad and the ugly from Run I

☛ The good

● Central computing systems were efficiently used – take the job to the data rather that the data to the job

● Resource sharing using batch systems and physics group disk allocations + common staging space worked well

● Primary datasets based on level 3 trigger decisions – stable against production versions

☛ The bad

● Bookkeeping for datasets, particularly those created by the physics groups left a lot to be desired. Home grown database – not very flexible

● Poor integration of desktop resources – no common login areas between central systems and desktops

☛ The ugly

● Analysis programs directly read tapes (mostly 8mm) – caused much pain

● Expansion of central UNIX analysis system relied on CPU upgrades – based on a single box

Overall Design Strategies

☛ The good

- Preserve the idea of resource sharing using batch system with common staging space + physics group disk

- Retain primary datasets creation based on level 3 triggers

- Retain concept of taking the job to the data

☛ The bad

- Improve robustness of bookkeeping by using a commercial database and including physics group datasets

- Improve integration of desktops – provide shared login areas with central system

☛ The ugly

- Eliminate direct reading of tape from user analysis job – spool data to disk first in a separate step

- Break dependence on a single SMP for central system

Datasets and data handling Strategy

- For efficient access to data on tape we must cluster according to expected use patterns

- The data will be organized by "datastream" where a datastream contains one or more datasets.

- Datasets are defined in Level 3, a dataset can consist of more than one Level 3 trigger. Datasets are inclusive, events can occur in more than one dataset.

- There will be about 50 datasets defined in Level 3.

- We do not want to write 50 RAW datastreams because of the operational complexity and cost of the staging disk required.

- We will organize the raw data into about 8 streams each containing about 6 datasets

- The grouping will be chosen to minimize event duplication and to keep similar sized streams together.

- The output of production will be PADs (Physics Analysis DST) of about 60KByte size. We will not keep the full output of reconstruction.

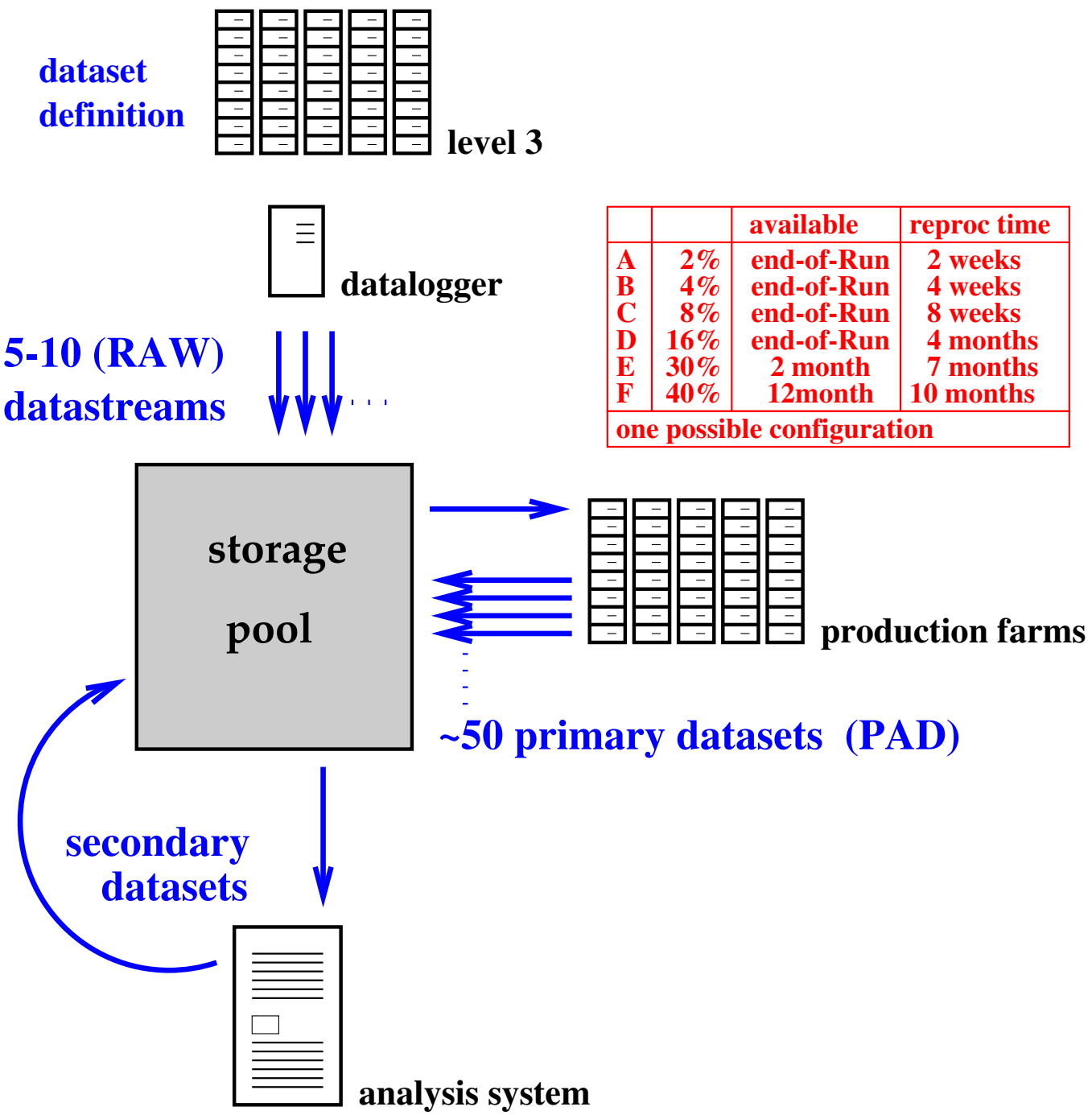
Strawman RAW datastreams

- Grouping based on expected size and predicted overlaps based on Run IB

Dataset	events ($2fb^{-1}$)	% of total
Stream 1		
$W \rightarrow e\ \nu$	3 E6	0.3
$W \rightarrow \mu\ \nu$	5 E6	0.5
$Z \rightarrow e\ e$	1 E6	0.1
$Z \rightarrow \mu\ \mu$	1 E6	0.1
High P_T e+jets	3 E6	0.3
High P_T μ+jets	3 E6	0.3
EM 18 GeV + tag	1 E5	0.01
Total	16.1 E6	1.61
Stream 2		
Photon 25 GeV, non-iso	2 E6	0.2
Photon 10 GeV, non-iso	3 E6	0.3
Jet 70 GeV	9 E6	1
Jet 50 GeV	9 E6	1
Jet 20 GeV	13 E6	1
Missing Et + jets	6 E6	0.6
Total	42 E6	4.1
Stream 3		
High P_T e	11 E6	1
High P_T mu	11 E6	1
High P_T plug e	3 E6	0.3
Top multijet	11 E6	1
Dijets, High Et	3 E6	0.3
Dilepton, SUSY	15 E6	2
EM 18 GeV + Missing E_T	4 E6	0.4
Total	58 E6	6
Stream 4		
$J/\Psi \rightarrow e\ e$	40 E6	4
Di-electron	50 E6	5
Di-muon	50 E6	5
$e - \mu$	50 E6	5
EM 18 GeV + mu	10 E6	1
2 EM 16 GeV non-iso	32 E6	3
2 EM 10 GeV isolated	10 E6	1
Total	242 E6	24

Dataset	events ($2fb^{-1}$)	% of total
Stream 5		
Low Pt e	100 E6	10
Low Pt mu	60 E6	6
$B \rightarrow\ \pi\ \pi$	70 E6	7
$B \rightarrow D_s\ \pi$	70 E6	7
Total	300 E6	30
Stream 6		
Photon + 2 Track	50 E6	5
Photon 50 GeV non-iso	22 E6	2
Photon 25 GeV isolated	58 E6	6
Jet 100 GeV	55 E6	6
Multijet	86 E6	9
Diffractive	42 E6	4
Low Missing Et + tag	40 E6	4
Total	353 E6	36
Stream 7		
$J/\Psi \rightarrow \mu\mu$	40 E6	
Stream 8		
Luminosity information	24 E6	

CDF Run II Dataflow, logical view



		available	reproc time
A	2%	end-of-Run	2 weeks
B	4%	end-of-Run	4 weeks
C	8%	end-of-Run	8 weeks
D	16%	end-of-Run	4 months
E	30%	2 month	7 months
F	40%	12month	10 months
one possible configuration			

- After running production the 8 RAW datastreams will be split into the 50 primary datasets.

- Primary datasets are defined by Level 3 trigger only

- Physicists will run selections on these datasets that reduce the number of events and/or the number of objects. We refer to these datasets as secondary and tertiary datasets

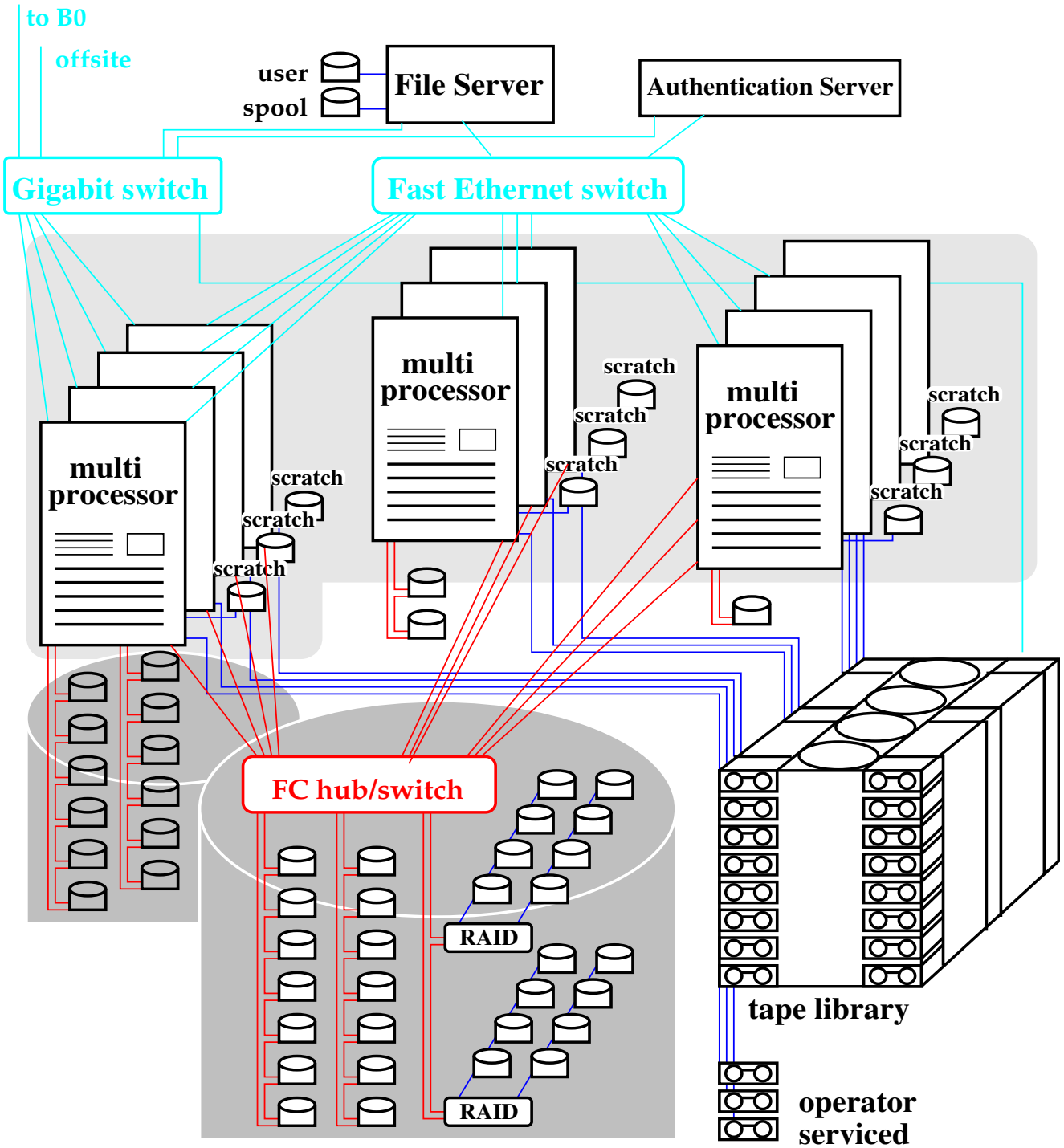
- Access to and cataloging of these datasets will also be through the CDF data access system.

ANALYSIS SYSTEM HARDWARE DESIGN

☛ Goals

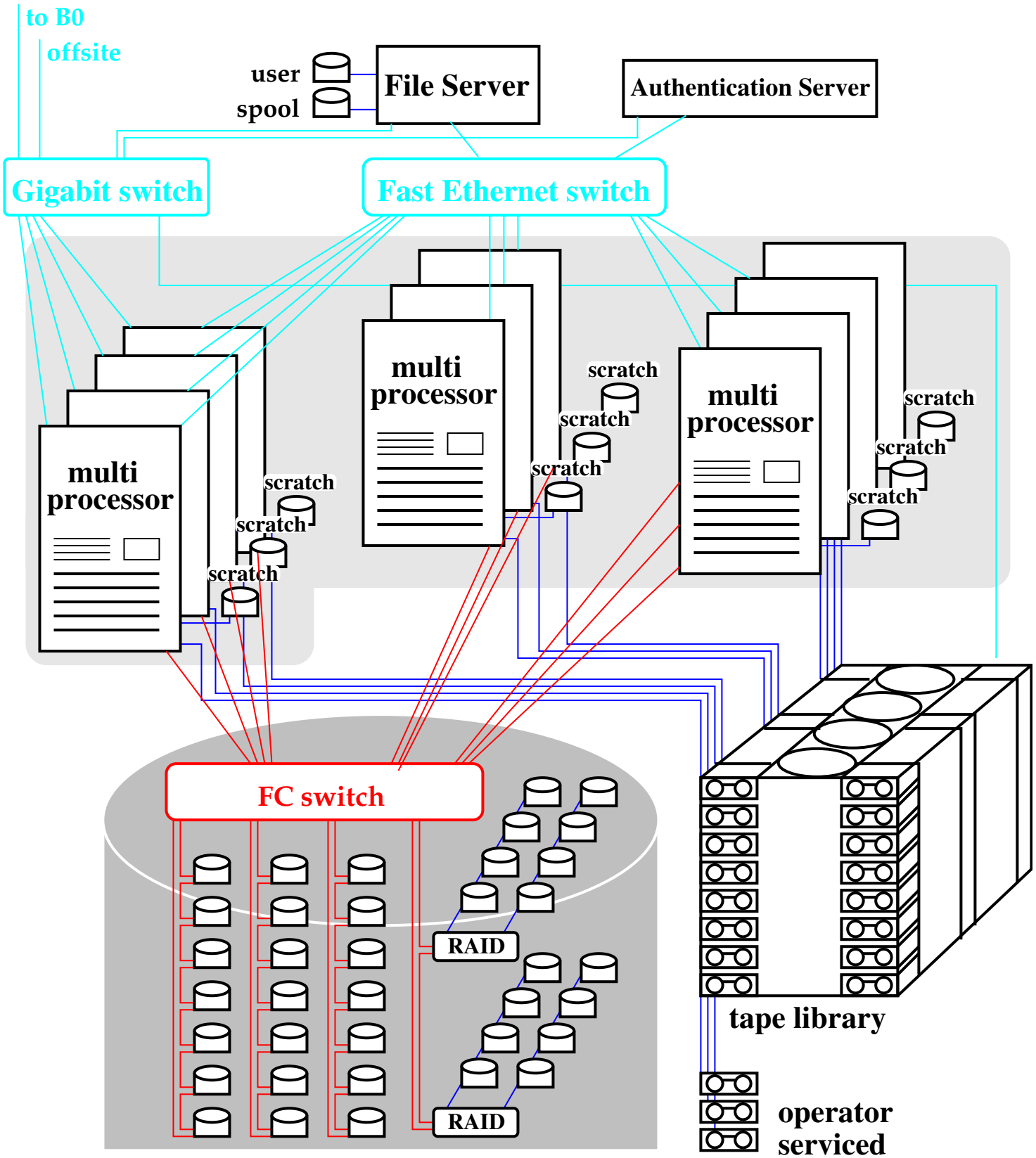
- System must be scalable beyond current CPU estimates (90,000 VUPS) at reasonable cost
- Put as much of the data as possible in robotic storage
- As much disk as we can afford (limited only by budget). Disk resident data makes the analysis possible
- Would like to be able to incorporate new technologies as they become available
- Better integration of desktop systems
- Continue to take the job to the data for efficient access

BASELINE DESIGN for ANALYSIS SYSTEM



Results of investigation of NFS server and Fibre Channel will be presented tomorrow by Stephan Lammel

ULTIMATE DESIGN



- The design was reviewed by an internal CDF committee in December

- The overall plan was endorsed by the committee

- They recommended:

  - ✪ Move more agressively with testing and deploying the fileserver to gain experience

  - ✪ Focus effort on one getting one system running and stable this year and build the cluster from there

DATA ACCESS SYSTEM DESIGN

- No global resource management – jobs scheduled individually based on available resources and job priority

- Eliminate direct tape access from analysis programs – spool to disk first

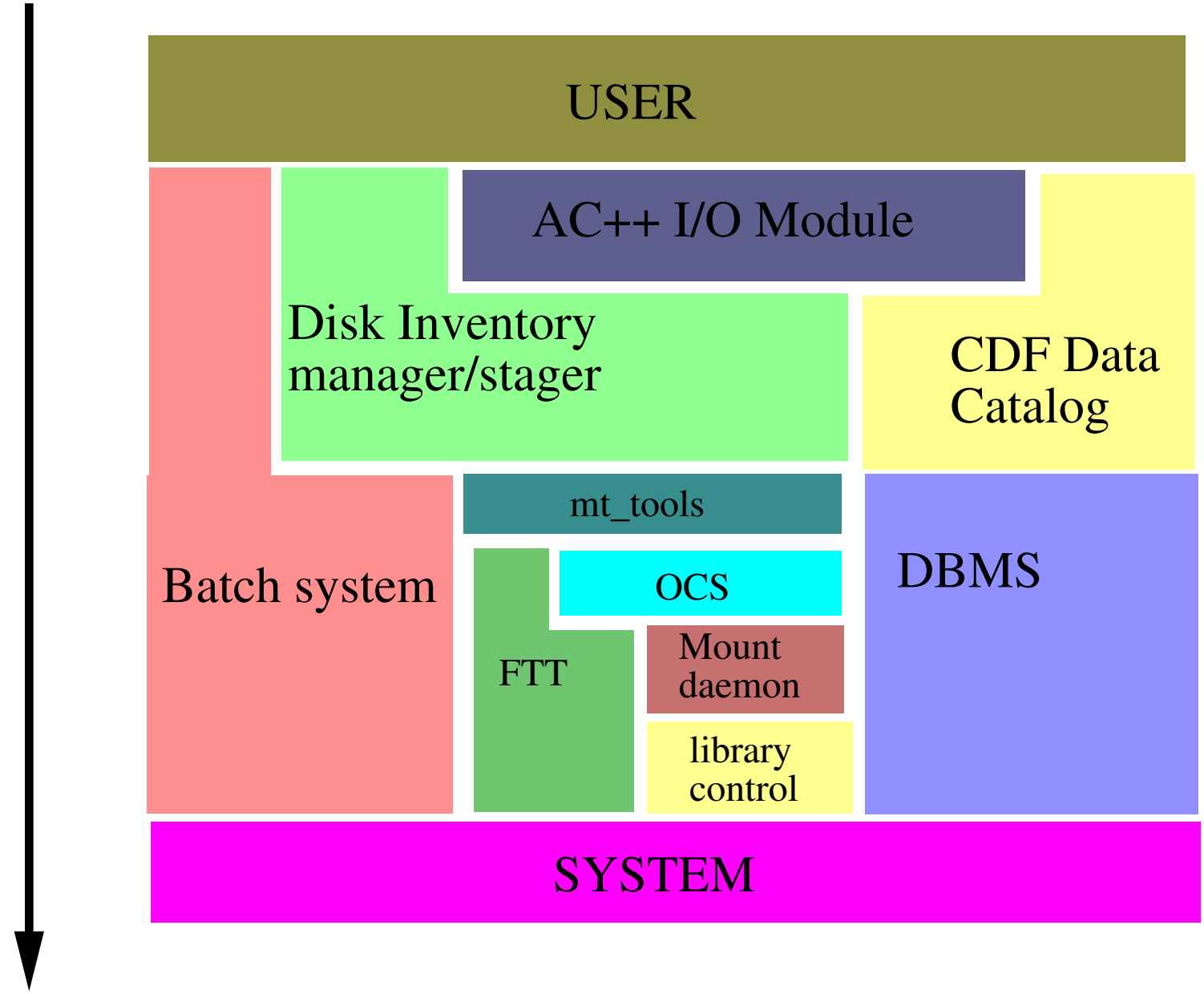
- Physics group specific resources – management and allocations delegated to physics group

- Maintain status information of an analysis job internal to the job – no outside intelligence involved

- Convenient access to any data in the system using the same interface – although some data may have large latency for retrieval

- Provide robust and easy–to–use bookkeeping to encourage cataloging of secondary and tertiary datasets and hence facilitate data sharing

COMPONENTS of CDF DATA ACCESS SYSTEM



Communication between components

CDF DATA CATALOG

- Use a robust professional DBMS, Oracle is the choice for use at FNAL. Probably require lightweight implementation for offsite use

- Catalog will be file–based, no event level information (a few million entries) but it will be possible to uniquely locate an event on a file/volume

- Organized into a hierarchy of tables

  Dataset → Fileset → File → Run Section

☛ Fileset

  ✪ Group of files located on the same partition on the same tape volume

  ✪ Partitions found in most modern tape technologies – can partition a physical tape into multiple logical tapes, can add and delete files within a partition – makes tapes less sequential in nature

  ✪ Some tape technologies allow load/unload at partition boundaries – potentially useful for event–pick mode.

☛ Run section

✪ Run is divided into sections for luminosity tracking purposes. Many thousands of run sections in a run. Will be needed to compute the luminosity of a dataset

● The catalog will contain pointers to production processing information, trigger table information, calibration constants used, run conditions, luminosity

● It will also maintain a list of free tapes available for allocation by online/production and users.

● A Web interface will be provided so that users can query the database to locate information.

● There will also be an API that is used by AC++ – this may be the same API (with necessary modifications) that is used for the calibration database.

☛ Status

✪ A prototype design exists, both in Oracle and msql with a first pass at the table design.

✪ We have loaded all the Run I bookkeeping records into it (~ 500,000)

✪ Pointers to other databases not yet defined

DISK INVENTORY MANAGER/STAGER

- This component is central to the whole design and is the one involving most work

- It provides the disk cache management and interface to the tape staging.

- It will manage all disk resident data except dedicated physics group disk space.

☞ Functions

✪ Keep track of name and location of all filesets that have been staged to disk

✪ Accepts reservations from users for use of a fileset and releases it when done.

✪ Allocates space to write new filesets

✪ It keeps track of whether a file is being spooled to tape, staged to disk, in use or permenantly disk resident

✪ Submits tape staging/spooling requests (will collect together all requests for files on the same tape from the same user – does not attempt to merge requests from different users)

- There is no communication with the CDF Data Catalog, all file/volume information is sent as part of the user request

- It will support files being stored on different tape technologies

- It supports the use of different tape staging mechanisms

- For use in a heterogeneous cluster it must know about filesystem formats and accessibility of disks on nodes

☛ Status

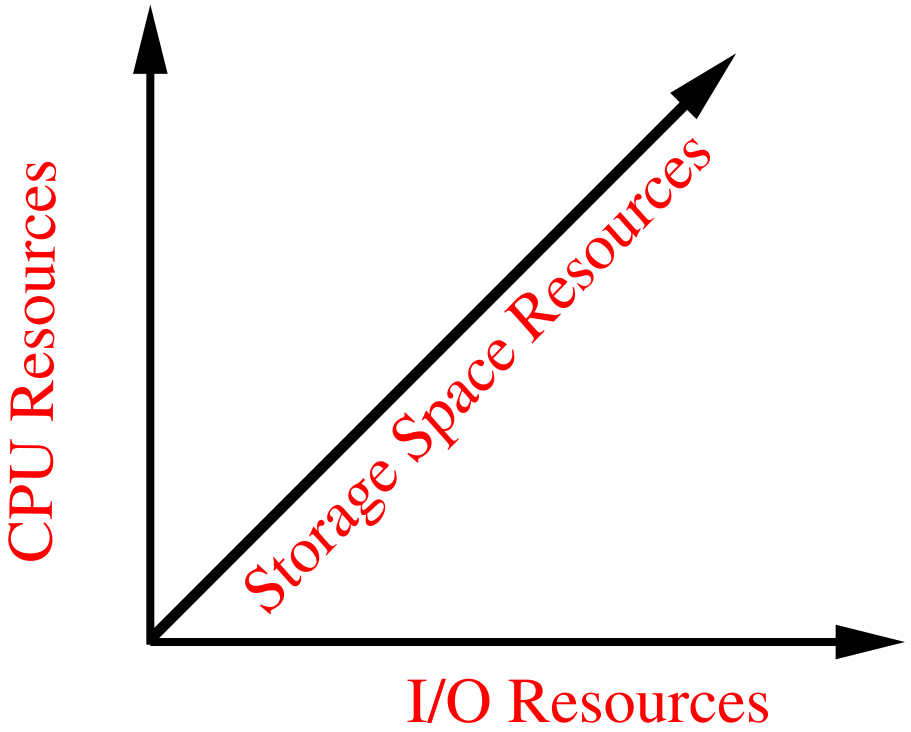
✪ A prototype exists and has been tested

✪ The design/tests of the prototype will be discussed in more detail tomorrow in the CDF Data Access talk by Terry Watts.

✪ We are hiring a consultant to help produce the full design

RESOURCE MANAGEMENT

- No Global resource optimization



- ☛ CPU Resources — manage through batch system
- ☛ I/O Resources — manage through batch system
- ☛ Storage space — staging space – DIM
  physics group space – static

☛ CPU Resources

● Need to manage better than in Run I. Used Sterling NQS on UNIX. Only had FCFS scheduling with no anti–flooding features. Complaints from users about people abusing the system. VMS batch was somewhat better. FNAL uses LSF on central systems but we never got around to installing it.

☛ I/O Resources

● Don’t need to manage disk I/O

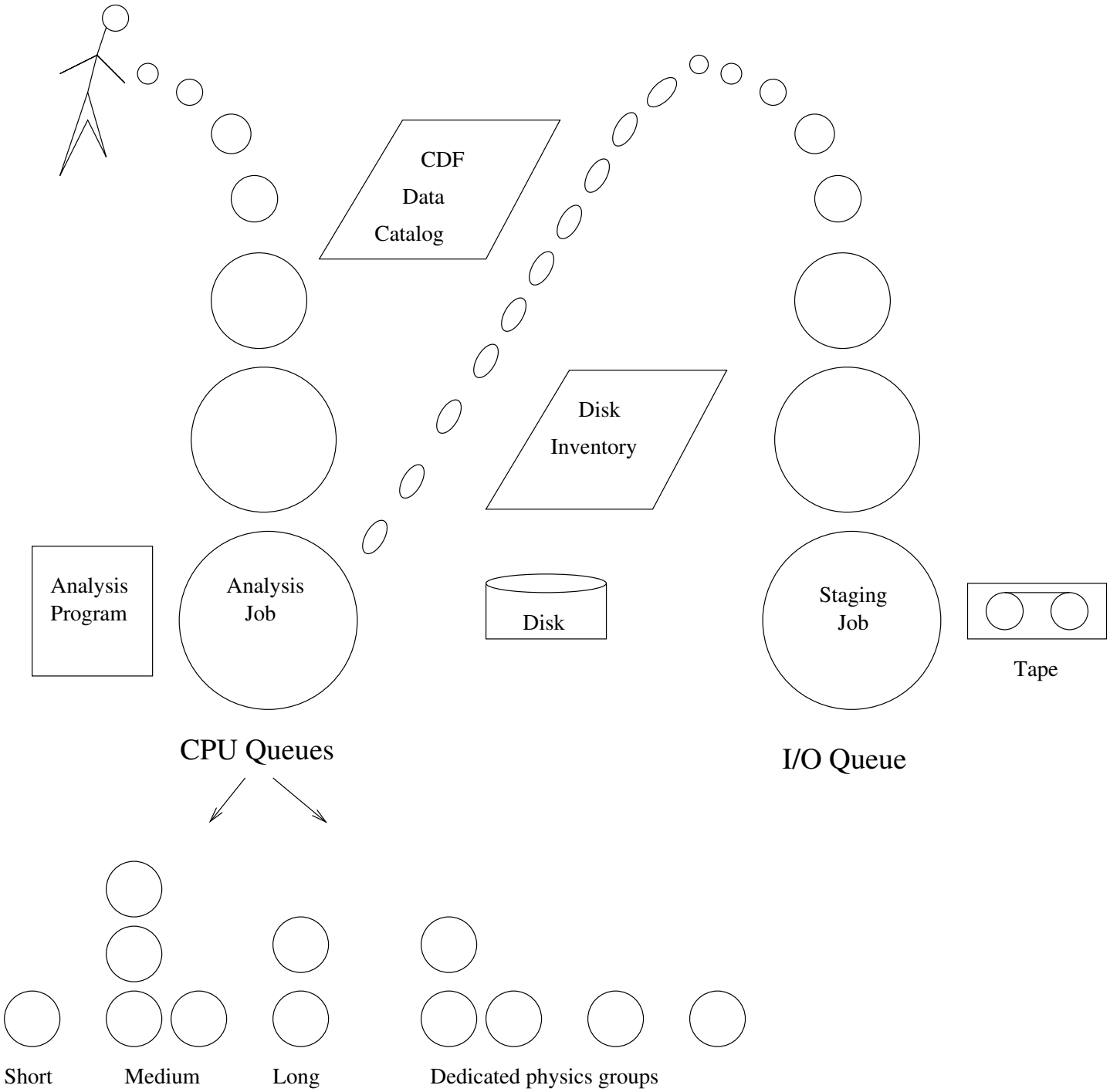
● Tape I/O and CPU needs are not really decoupled. Job requires a mixture of both at the same time on the same machine.

● We will however attempt to manage them separately.

● This is possible because the analysis job will not read tape directly

● Multiple users can access the same data on the same tape simultaneously – one user mounts the tape and the others read from disk

- Both these resources will be managed using a batch system with a set of queues for analysis jobs based on cpu requirements and queues for tape input and output



- Developing a full list of requirements for the batch system in conjunction with the Joint Batch Working Group

☛ Some important features

- Fairshare scheduling based on priority of analysis, current and past cpu consumption, amount of cpu requested, number of current running jobs of user/group

- Fault tolerant – another node in the analysis cluster should take over if the master fails.

- Ability to move the users' files from the fileserver to the execution node – removes dependence on network especially for long jobs

- Ability to assign different priorities to different queues

- Prevent flooding of system by a single user

☛ Evaluation of LSF (Platform Computing) against the requirements in ongoing. LSF has been used in the prototype and the current indications are that it will meet our needs with appropriate customizations.

☛ Storage Space Resources

User areas	quotas + self–management by user
Scratch space	quotas + scratch daemon
Staging space	dynamic using DIM. Users will be encouraged to inventory datasets so they are available through DIM Fix frequently used data on disk
Physics group disk	static Managed through physics groups
Tape space	static, through administrator User allocated tape pool

☛ Robot Arm time

Plan to read maximum amount of data/tape mount – full tapes if possible

Given the capacity of the robot (256 drives) we don't believe we have to manage the robot arm time

Interface to AC++ – I/O Modules

- Analysis jobs will be run using the standard AC++ framework

- AC++ allows the inclusion of custom input/output modules that read and write data

☞ Input Module

- We would like users to think in terms of analysing datasets whenever possible as this will be the most efficient way to access large amounts of data

- The input module will use the CDF data catalog to translate the dataset into a list of files.

- File will be processed in the optimal order, those that are already on disk will be processed before those on tape – in Run I the files were always processed in the order specified.

- Input module to process files and lists of files will still be available if the order of processing must be specified.

- The design is mostly understood. More details in Terry's talk tomorrow.

☛ Output module

- The output module will write files to disk which will then be spooled to tape in a separate step. Space will be allocated for writing by the DIM.

- It will record new files in the CDF catalog

- In addition it will have an automatic file close/open mode for the online, production and dataset creation to write files close to a target filesize of 1 Gbyte (large files are necessary to make good use of the tape drives).

- It will also sort events at run section boundaries so that all events belonging to one run section and dataset/datastream are in the same file. This avoids the need to the event level entries in the CDF catalog.

- More work is needed on the design of the output module

TAPE STORAGE INTERFACE

- Management of the tape I/O resources will be controlled by the policies set in the batch queues and all file/volume information is stored in the CDF catalog.

- In the prototype the access to the EMASS robot was done using FTT and OCS along with a thin software layer mt_tools that writes ANSI labelled tapes and has error recovery and retry. The mt_tools package was used extensively in Run I to spool data to and from tape (~100 TBytes).

- The baseline design for the hardware is to use directly attached tapes on the analysis systems. As long as this remains the case, our tape handling solution is appropriate.

- We expect to export the data handling system to university groups so we need to have a tape interface that we can export to our offsite users.

- The DIM is designed to support the use of other tape staging mechanisms so the use of ENSTORE is not precluded should CDF chose to move to network attached tape drives

Production Farms

- Production will be done on farms of PCs running Linux

- The design of the farms (hardware and software) is the responsibility of the Joint Project on Production Management.

- The cpu estimates for CDF processing range between 48,000 to 72,000 VUPS. Additional needs such as Monte Carlo production and reprocessing of selected datasets could double this number.

- We expect to need one or more I/O nodes to handle the splitting into streams and the spooling to tape in the robot. The farms will use the same tape spooling system as the data handling and will have dedicated tape drives in the robot directly attached to the I/O nodes.

- The farm worker nodes will process the files one of the 8 RAW datatreams and send the output to the I/O node where it will be split in into the individual datasets.

- In order to reach the target filesize of 1 GByte we will have to combine these split files together into large files.

- Files will be accumulated on the I/O node until there are enough to combine and then a new ouput file will be created.

- The intermediate files are not cataloged and will be discarded once the combination step is successful. The combined files will be cataloged in the CDF catalog

- Data will be spooled to tape once there is sufficient data of a particular stream to fill a tape. If we have very small streams we may need to append to tapes.

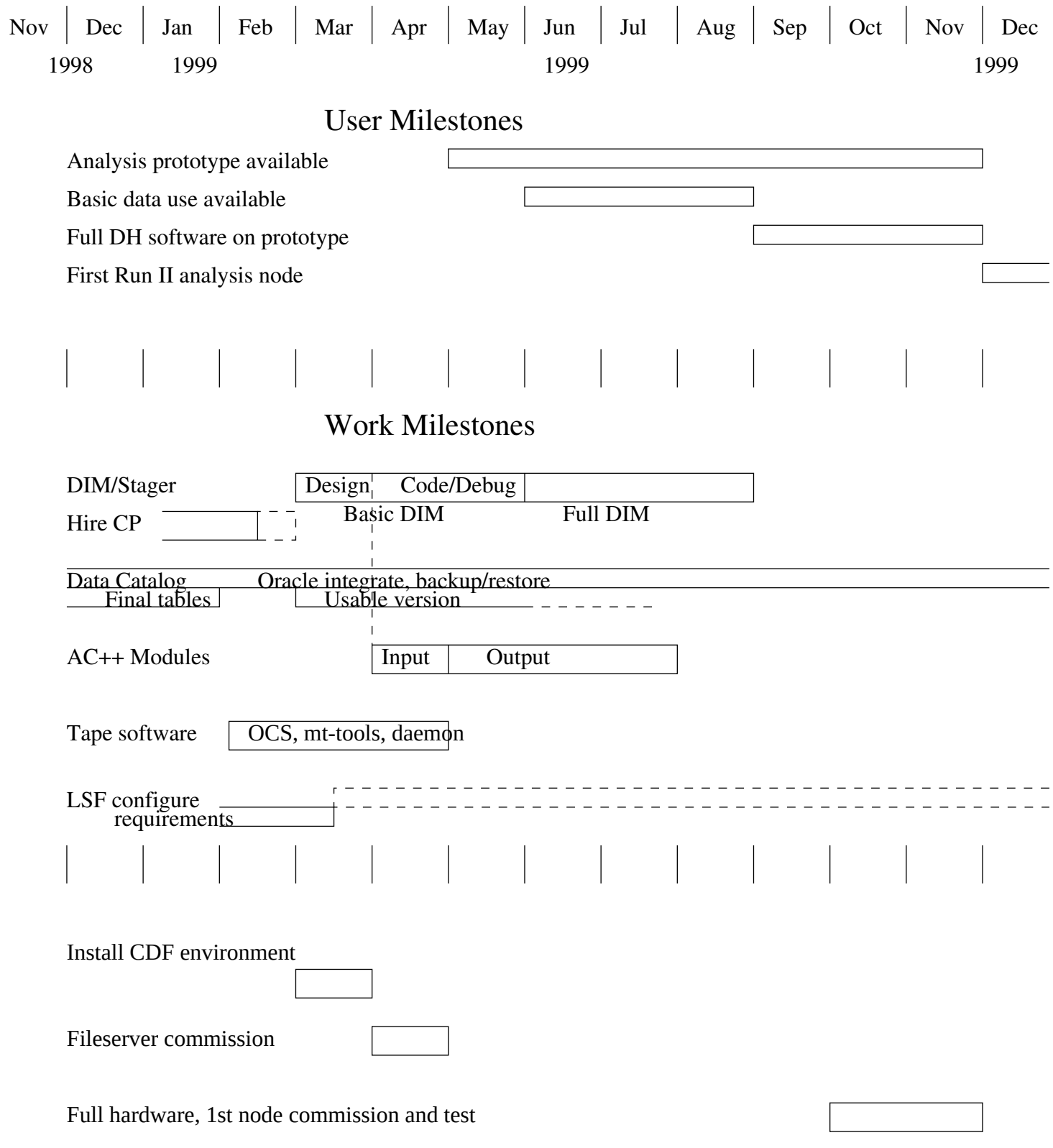
- We have not decided whether we will keep an unsplit inclusive PAD file for each RAW datastream input file. It takes more storage but could be very useful in recovering damaged tapes containing split data without having to reprocess the RAW data.

☛ Status

✪ The current CDF production executable is running on the prototype farm (setup as part of the Joint Project).

✪ The CDF farms group is evaluating the Farms Batch System (FBS) and understanding how to configure and control the flow of data through the farm

SCHEDULE and MANPOWER For CDF Data Access Joint Project



☛ Current CDF JP Data Access manpower

Person	Task	Fraction
Stephan Lammel(Co–ldr)	overall design	0.5
Terry Watts(Co–ldr)	Prototype tests	
Liz Buckley–Geer	Batch	0.3
Eric Wicklund	Data catalog	0.5

! In order to meet the schedule outlined we need the following additional manpower

1FTE	DIM	Hire consultant
0.5FTE	Batch integration	Postdoc or CP
0.5FTE	DIM integration	Postdoc or CP
0.5 FTE	Data catalog	CP
0.5FTE	Batch customization	CP

! We have already slipped 4 months on the start date for developing the complete disk inventory manager due to lack of manpower

! To keep the current schedule for the DIM (which drives many other things) we must hire the consultant by mid–February. We are in the process of interviewing candidates.

- The batch customization CP will come from HPPC/Farms group once the Batch Working group has determined the needs

- The CP for data catalog will come from the Support Databases joint project.

- We need the 2 additional postdocs/CPs very soon if we are to maintain the current schedule

Conclusions

- We have a good, well thought out design for the CDF data access system and analysis hardware that will meet the needs of the experiment for Run II.

- The design draws on our experiences in Run I, keeping the good while fixing the bad.

However:

- We will not be ready in time if we do not get the necessary additional manpower very soon.

- We have already slipped 4 months on the start date for developing the complete disk inventory manager due to lack of manpower